

---

# CELLULAR STATE TRANSFORMATIONS USING GENERATIVE ADVERSARIAL NETWORKS

---

A PREPRINT

**Colin Targonski**

Department of Electrical and Computer Engineering  
Clemson University  
Clemson, SC 29634  
ctargon@clemson.edu

**Benjamin T. Shealy**

Department of Electrical and Computer Engineering  
Clemson University  
Clemson, SC 29634  
btsheal@clemson.edu

**Melissa C. Smith**

Department of Electrical and Computer Engineering  
Clemson University  
Clemson, SC 29634  
smithmc@clemson.edu

**F. Alex Feltus\***

Department of Genetics & Biochemistry  
Clemson University  
Clemson, SC 29634  
ffeltus@clemson.edu

July 2, 2019

## ABSTRACT

We introduce a novel method to unite deep learning with biology by which generative adversarial networks (GANs) generate transcriptome perturbations and reveal condition-defining gene expression patterns. We find that a generator conditioned to perturb any input gene expression profile simulates a realistic transition between source and target RNA expression states. The perturbed samples follow a similar distribution to original samples from the dataset, also suggesting these are biologically meaningful perturbations. Finally, we show that it is possible to identify the genes most positively and negatively perturbed by the generator and that the enriched biological function of the perturbed genes are realistic. We call the framework the Transcriptome State Perturbation Generator (TSPG), which is open source software available at <https://github.com/ctargon/TSPG>.

**Keywords** Generative Adversarial Networks · Bioinformatics · Adversarial Examples

## 1 Introduction

RNA sequencing (RNAseq) uses the capabilities of high-throughput DNA sequencing methods to provide insight into the transcriptional state of a biological sample [1]. An RNAseq gene expression profile can be coupled with metadata describing the sample, thus placing it in a biological context such as tissue source, aberrant phenotype such as disease, and other variables such as age, sex, medical history, and environmental factors. Traditional methods of high dimensional analysis are challenging to use on RNAseq datasets because of the high feature to sample ratio and often high noise levels in biological systems. Modern machine learning and deep learning approaches have begun to surmount these obstacles and are increasingly used to cluster and classify gene expression profiles into meaningful groups that correspond with metadata labels [2, 3, 4].

Deep learning [5] has had tremendous success in image [6, 7] and natural language [8, 9] processing tasks due to its ability to abstract high level features from highly dimensional and noisy datasets. Deep learning has also been incorporated as the underlying framework behind powerful generative models such as variational autoencoders (VAEs) [10] and generative adversarial networks (GANs) [11]. GANs, a focus of this work, consist of a generator that captures a data distribution and a discriminator that estimates the probability that a sample came from the training data rather

---

\*Corresponding author.

than the generator. As training progresses, the generator produces increasingly realistic data while the discriminator becomes more adept at distinguishing real from fake. GANs have surged in popularity in the computer vision field due to several works creating exceptionally realistic images [12, 13, 14].

While deep learning has proven itself useful in a variety of fields, surprising vulnerabilities exist within trained models. Adversarial examples are inputs to neural networks that an attacker has designed with the intention to cause the model to make a mistake [15, 16]. The general anatomy of an adversarial example consists of a calculated perturbation that is added to an input, which is subsequently passed to a trained model. Adversarial examples are generated to "trick" a neural network into confidently choosing a target class. For example, image  $x$ , classified as a horse by a neural network, can be subtly perturbed and classified by the same neural network as a human. Previous works [16, 17, 18] have led to Xiao *et al* [19] proposing AdvGAN, a framework to produce perceptually realistic images in an efficient manner using GANs. In this report, we adapt the AdvGAN framework to model the transition between tissue transcriptome states.

Within the field of molecular biology, a limited amount of work involving generative models has been proposed. Ghahramani *et al* [20] proposed a Wasserstein GAN [21] to integrate epidermal datasets by generating samples that cover the full diversity of cell types. Additionally, the authors use the generative model for both dimensionality reduction and to observe the effect of cell state perturbations on gene expression. Dizaji *et al* [22] introduced a semi-supervised approach to generate gene expression profiles of target genes using landmark genes and GANs. This work adapts the AdvGAN framework to analyze gene expression patterns during cellular state transformations by training a generator on a specific set of genes with a conditional target.

## 2 Methods

### 2.1 Datasets and Dataset Preparation

This study uses two gene expression matrices (GEMs) containing RNAseq expression levels for human tissue and tumor samples. The first dataset tested is the Genotype-Tissue Expression (GTEx) dataset, which contains 11,688 samples across 56,202 genes representing 53 uniquely labeled tissue types. The second dataset considered is the Cancer Genome Atlas (TCGA) dataset [23], which contains 11,092 patients across 60,483 genes representing 33 unique tumor types of distinct disease progression. We normalize expression levels by scaling each feature (gene) between the range  $[0, 1]$ . We use the Broad Institute Molecular Signature Database gene subsets (MSigDB v6.2) (MSigDB) [24] to extract sub-GEMs from GTEx and TCGA of lower dimensionality in order to reduce genetic complexity and obtain more interpretable results. We normalize the GEM data using the Python sci-kit learn package [25]. All neural network development and training is performed using Tensorflow [26]. Training of the networks utilized the Clemson University Palmetto Cluster using Nvidia V100 GPUs. The entire implementation is available on GitHub at <https://github.com/ctargon/TSPG>.

### 2.2 Transcriptome State Perturbation Generator (TSPG)

The general framework of the GAN used in this work was adapted from the AdvGAN model introduced by [19], which was influenced and derived from previous models and criterion metrics from both the GAN research community and the adversarial attack community. To formulate the adversarial generation that TSPG aims to perform, let us consider an input pair  $(x_i, y_i)$  to be the  $i$ th input pair of a dataset consisting of feature vector  $x_i \in X$  generated according to an unknown distribution  $x_i \sim P_{data}$ , where  $X \subseteq \mathbb{R}^n$  with  $n$  being the number of genes (features), and  $y_i \in Y$  the corresponding label. The goal of a classification system, which we will refer to as the target model, is to learn a mapping  $f : X \rightarrow Y$  from input domain to label domain. The goal of the adversary (in this case, the generator) is to learn to produce  $x_{adv}$  that is classified as  $f(x_{adv}) = t$  where  $t$  is a target (the targeted case) and different from the ground truth label  $y$ .

The TSPG framework can be seen in Figure 1 and contains a generator  $G$ , a discriminator  $D$ , and a pretrained target network  $f$ .  $G$  takes an input vector  $x$ , with the number of genes being the number of dimensions, and generates a perturbation  $G(x)$ .  $G(x)$  is added to the original instance to create an adversarial example  $x_{adv} = x + G(x)$  which is then passed to networks  $D$  and  $f$ . The goal of  $D$  is to encourage the generated example  $x_{adv}$  to be indistinguishable from data sampled from  $P_{data}$ , which is quantified by  $L_{GAN}$  and shown below:

$$L_{GAN} = \mathbb{E}_x[\log D(x)] - \mathbb{E}_x[\log(1 - D(x + G(x)))] \quad (1)$$

For  $L_{GAN}$ , we use the least squares objective proposed in [27], as it has been shown to stabilize training and boost results. The network  $f$  outputs a class prediction for input  $x_{adv}$  based on the pretrained weights. The goal of the generator is to produce adversarial example  $x_{adv}$  that is classified as  $t$  by  $f$ . The loss for tricking  $f$  in such a manner, proposed by [17], is:

$$L_{adv}^f = \max(\max_{i \neq t} f(x_{adv})_i - f(x_{adv})_t, \kappa) \quad (2)$$

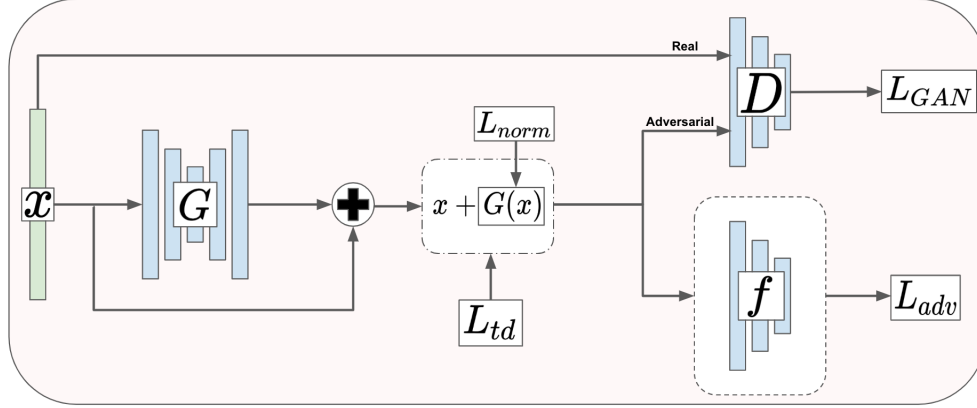


Figure 1: Architecture of transcriptome state perturbation generator (TSPG).

where  $t$  is the target class, and we set confidence  $\kappa = 0$ . This loss term encourages the generator to produce data that fools  $f$  into predicting  $t$ .

We also use two terms,  $L_{norm}$  and  $L_{td}$ , which are used to stabilize training and encourage  $x_{adv}$  to fall within the target data distribution  $P_t$ .  $L_{norm}$  is simply the  $L_2$  norm placed on the generated perturbation  $G(x)$ :

$$L_{norm} = \|G(x)\|_2 \quad (3)$$

This loss encourages the overall perturbation to be smaller, and we found this stabilizes training and produces better results. The final criterion for the loss term,  $L_{td}$ , is a measure of  $L_1$  distance between an adversary example  $x_{adv}$  and a randomly sampled target vector  $r_t$ , where  $r_t \sim \mathcal{N}(\mu_t, \Sigma_t)$ .  $\mathcal{N}(\mu_t, \Sigma_t)$  is a normal distribution modeled with the mean genetic expression level  $\mu$  of target class  $t$  and covariance matrix  $\Sigma$  of target class  $t$ . The loss is formulated as:

$$L_{td} = |x_{adv} - r_t| \quad (4)$$

$L_{td}$  encourages the generator to create data more tightly within the bounds of the target class distribution. Similar principles are used in [12, 28] to reduce the space of possible mappings and create sharper data instances. Together, the loss for  $G$  can be summed as:

$$L = L_{GAN} + L_{adv} + L_{norm} + L_{td} \quad (5)$$

The architectures used by  $G$  and  $D$  are similar to those used in [19], which are influenced from image to image translation architectures [12, 28].  $G$  adopts an hourglass shaped structure which contains three downsampling fully connected layers of sizes 512, 256, and 128, followed by three residual blocks, finishing with two upsampling layers of sizes 256 and 512 and then a layer of the same shape as the input dimension. The rectified linear unit (ReLU) activation function is used on each layer barring the final output, which uses the hyperbolic tangent (tanh). Each hidden dense layer is coupled with a batch normalization layer using default Tensorflow parameters. The tanh function allows the generator to produce a perturbation ranging from  $[-1, 1]$ . After the  $x_{adv} = x + G(x)$  operation,  $x_{adv}$  is clipped to values between  $[0, 1]$ , which are valid gene expression values.

The discriminator uses a small architecture containing three dense layers of units 512, 256, and 128. Each dense layer is followed by a batch normalization layer and a leaky ReLU activation function with alpha set to 0.2. The target model used in the experiments contains three dense layers of size 1024, 512, and 128, each layer using the ReLU activation function. All three networks are trained using the Adam optimizer [29] with learning rates 0.0002, 0.0001, and 0.001 for the generator, discriminator, and target model, respectively. The generator uses bootstrapping by randomly selecting 80% of each class for training and holding out the remaining 20% for testing.

### 2.3 Data Visualization

Once a generator is trained using the framework and a given target, an input vector was fed to the network for inference, resulting in a perturbation and newly spawned adversarial example. We chose to perturb 100 randomly chosen samples from a given GEM dataset, and then we plotted the results using t-SNE [30] along with the samples from 10 randomly selected classes, including the target class. This created a low dimensional representation of the sample distributions, allowing us to observe what distribution the adversarial examples follow. We also provide methods to view the RNAseq changes in the form of a heatmap. The components involved in a targeted perturbation generation include a sample, a

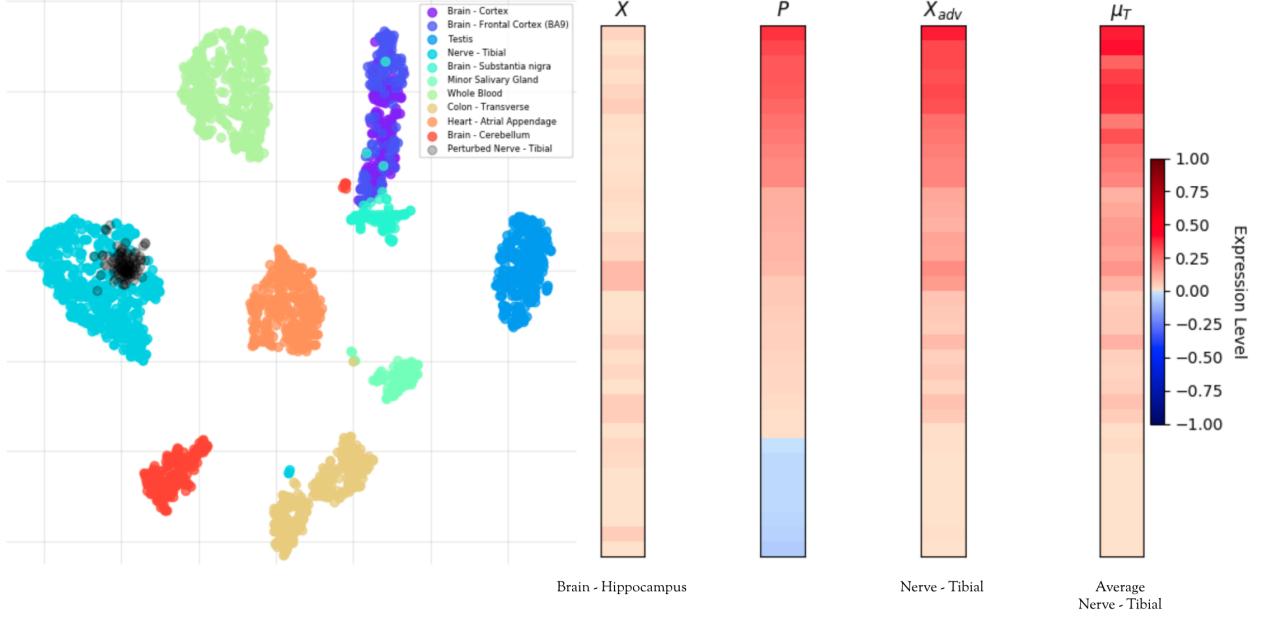


Figure 2: **Adversarial generation for Nerve-Tibial target using the Hallmark Hedgehog Signaling gene set.** t-SNE plot of original and perturbed samples using the Hallmark Hedgehog Signaling gene set (left). Heatmap of cellular transformation from Brain-Hippocampus to Nerve-Tibial (right). Perturbation ( $P$ ) ranges from  $[-1, 1]$ , which is added to original sample ( $X$ ), then adversarial example ( $x_{adv}$ ) is clipped to  $[0, 1]$ . The mean expression vector ( $\mu_T$ ) of the target class (Nerve-Tibial) is shown.

perturbation, and an adversarial example. Additionally, we used the mean vector of a target class to visually inspect how similar an adversarial example is to the target class. With these ingredients, we can produce a plot depicting a gene-wise transformation of an input cell to a target cell, including the RNAseq expression level differences.

## 2.4 Gene Set Functional Profiling

The top and bottom twenty most highly perturbed genes were input into the ToppFun tool (<https://toppgene.cchmc.org>) [31] in March 2019. Occasionally, the HGNC symbol was not recognized for 1/20 genes in the list. Functional labels were considered significant at a q-value Benjamini and Hochberg corrected False Discovery Rate of  $q < 0.0001$ .

## 3 Results

We first examined the raw adversarial generation capabilities using 50 "Hallmark" gene sets defined in MSigDB. Table 1 contains the results for various Hallmark gene sets being used as feature input to the TSPG framework, as well as the corresponding target class. The furthest right column,  $f$  Accuracy, represents the target model accuracy when the input is perturbed *from any input source* (i.e. all samples from the dataset) and the label is the target class. Thus, the accuracy represents the ability of the generator to "trick" the target model into classifying a sample as a specified class.

Table 1: Accuracy of target network on perturbed datasets.

Gene Set	Num Genes	Target Class	$f$ Accuracy
Hallmark Hedgehog Signaling	36	Nerve - Tibial	100%
Hallmark Peroxisome	107	Brain - Spinal Chord	100%
Hallmark Apoptosis	161	Lung	99.9%
Hallmark E2F Targets	200	Artery - Coronary	99.8%
Hallmark All	4386	Thyroid	100%
Hallmark All	4386	Heart - Left Ventricle	100%

The high classification accuracy suggests that the generator is powerful enough to exploit the decision boundary the target model defined no matter what class the input came from. While this is in itself interesting, we still did not know what distribution the adversarial samples might follow. This left us with three questions: 1) Will generated adversarial examples follow the distribution of the target class? 2) Are the perturbations created by the generator nonsensically changing expression values to satisfy the criterion of tricking a target model? 3) Do the most highly perturbed genes encode biological function of the target condition?

To test if the adversarial examples exhibited similar high dimensional structure as the original target samples, we used the dimensionality reduction and visualization tool t-SNE [30]. This technique allowed us to ask if perturbed samples clustered with the target condition distribution. In Figure 2, ten different classes from the GTEx dataset were plotted, as well as 100 randomly chosen adversarial examples that were created from varying source tissues in the original dataset. Each sample was originally 36 dimensions, representing the 36 genes in the Hallmark Hedgehog Signaling gene set. We observed that the original (unperturbed) Nerve-Tibial samples (dark cyan) and the adversarial (perturbed) Nerve-Tibial samples (black) formed a cluster on the left side of the plot. The adversarial (black) points were originally 100 different samples from the GTEx dataset of varying class description, then were passed through the generative process as described in Figure 1, thus transforming these points to the target condition. It is evident from Figure 2 that the perturbed samples follow a similar data distribution as the original target condition.

The right hand side of Figure 2 shows the step by step process of generating an adversarial example. In this case,  $x$  was an original sample from the GTEx dataset of class type Brain-Hippocampus.  $x$  was passed through the generator to produce  $P$ , which was then added to  $x$  to create adversarial example  $x_{adv}$ .  $P$  ranges between  $[-1, 1]$  to allow the generator to entirely "activate" or "silence" any gene.  $x_{adv}$  was clipped to valid expression values in the range of  $[0, 1]$ . This generated example  $x_{adv}$  was then classified as target class  $t$  by a pretrained model (the target model). What was left are the perturbation values  $P$  describing the exact gene expression changes, on a per gene basis, that occurred in order to transform, in this instance, Brain-Hippocampus tissue to Nerve-Tibial tissue. Additionally, we showed  $\mu_t$ , the mean expression vector for the target class (Nerve-Tibial in this example) for direct visual comparison between an adversarial example  $x_{adv}$  and the target class.

We also include Figures 3 and 4 as examples of TSPG working with a larger input gene set where all 4,386 Hallmark genes were combined. In this case, the perturbed samples clustered with original samples of both target classes (Heart - Left Ventricle and Muscle-Skeletal). Additionally, if we visually inspect the similarity between  $x_{adv}$  and  $\mu_T$  in each heatmap, the expression vectors appear to be similar. This experiment demonstrates the ability of TSPG to generalize to variably sized input gene sets.

The classification accuracy, t-SNE visualizations, and perturbation heatmap results suggest that the perturbation of the hallmark gene expression patterns shift the state of the source tissue to the target tissue. To test if the perturbed genes are at all relevant to the source or target tissue, we randomly selected a sample from each of the 53 tissues available in the GTEx dataset and passed it through the generator that was trained with Heart, Left Ventricle being the target. Then, we took the 20 most *positively perturbed* genes (genes being "turned on" or expressed) and the 20 most *negatively perturbed* genes (genes being "turned off" or silenced) for each of the 53 state transitions. Interestingly, only 85 out of 4,386 genes were found in the top 20 most perturbed list for each of the 53 GTEx experiments suggesting that the same genes tend to be perturbed towards the target tissue. On the other hand, the twenty most negatively perturbed genes summed to 642 unique genes suggesting that a greater variety of genes must be "turned off" depending on the phenotype of the source tissues.

We next asked if the 20 most positively perturbed genes out of the pool of 4,386 genes were enriched (B&H FDR  $q < 0.0001$ ) for biological function relevant to the target tissue (i.e. Heart-Left Ventricle). A representative enriched term for each of the 16 term categories and the total number of enriched terms across all 53 experiments is shown in Table 2. Clearly, many of the enriched terms are related to muscle and heart biology. Conversely, we asked if the 20 most negatively perturbed genes were related to the source tissue's biology. Supplemental Table S1 shows the most significant PubMed scientific articles that were non-randomly associated with these genes. We found that the source tissue is related to the PubMed articles associated with these genes in most cases. All term enrichments can be explored in Supplemental Table S2.

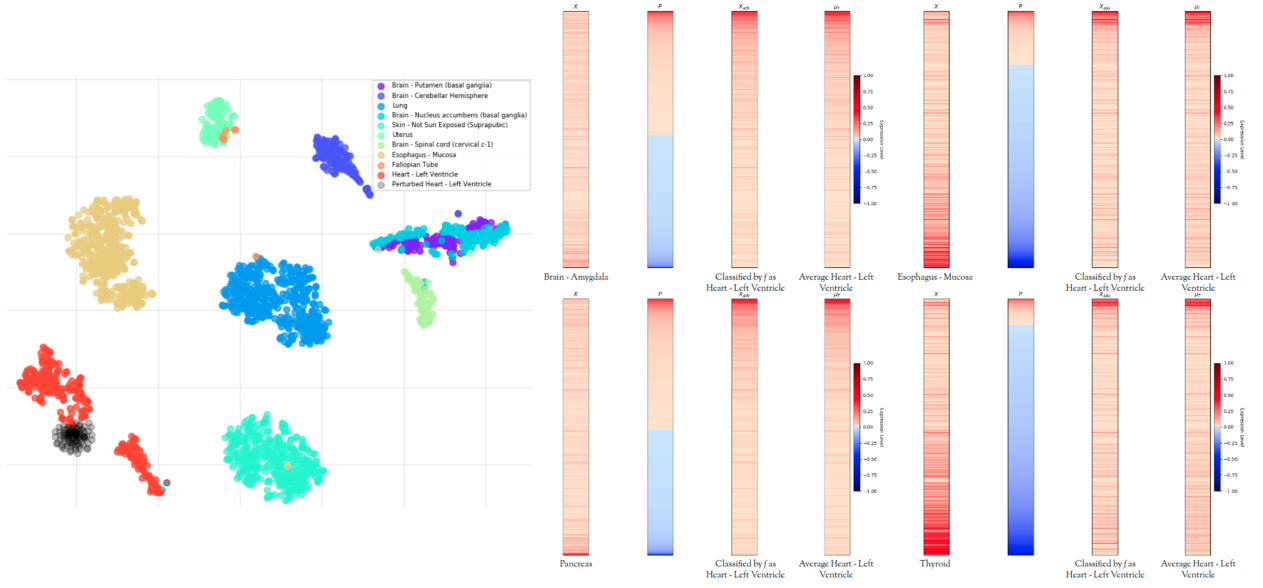


Figure 3: **Adversarial generation for Heart-Left Ventricle target using all Hallmark genes as the input gene set.** t-SNE plot of original and perturbed samples using the all Hallmark genes (left). Heatmap of cellular transformations from Brain-Amygdala, Esophagus-Mucosa, Pancreas, and Thyroid to to Heart-Left Ventricle (right). Perturbations ( $P$ ) range from  $[-1, 1]$ , which is added to original sample ( $x$ ), then adversarial example ( $x_{adv}$ ) is clipped to  $[0, 1]$ . The mean expression vector ( $\mu_T$ ) of the target class (Heart-Left Ventricle) is shown.

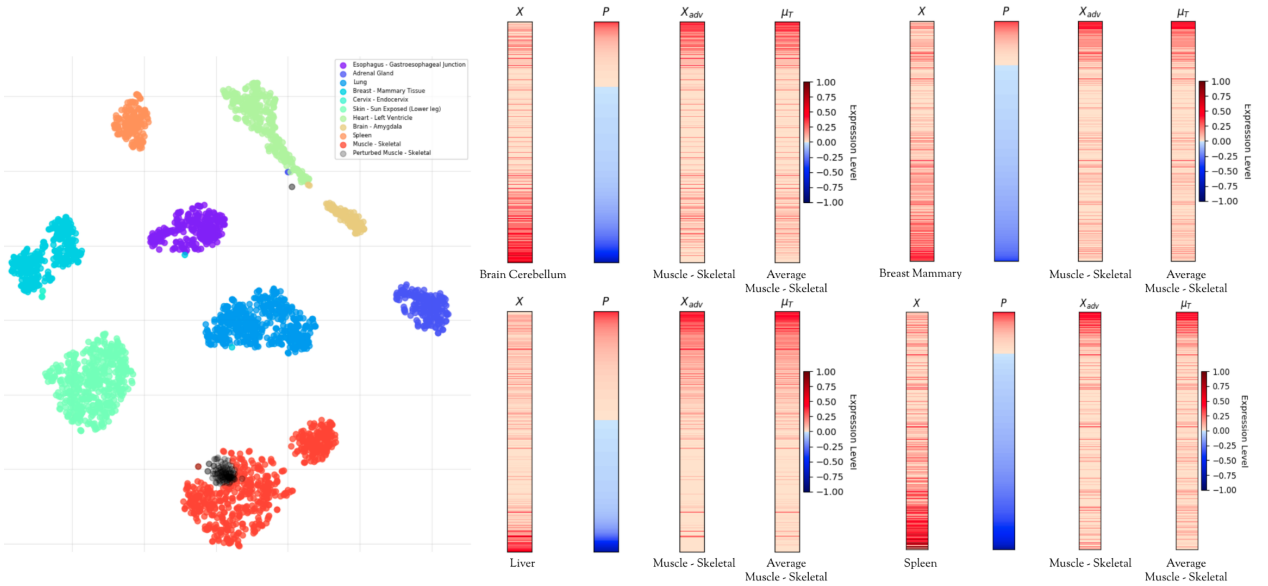


Figure 4: **Adversarial generation for Muscle-Skeletal target using all Hallmark genes as the input gene set.** t-SNE plot of original and perturbed samples using the all Hallmark genes (left). Heatmap of cellular transformations from Brain-Cerebellum, Breast-Mammary, Liver, and Spleen to to Muscle-Skeletal (right). Perturbations ( $P$ ) range from  $[-1, 1]$ , which is added to original sample ( $x$ ), then adversarial example ( $x_{adv}$ ) is clipped to  $[0, 1]$ . The mean expression vector ( $\mu_T$ ) of the target class (Muscle-Skeletal) is shown.

Table 2: Enriched Biological Functions of the Top Twenty Most Positively Perturbed Genes Leading to the Target Condition Heart-Left Ventricle

Term Category	Enriched Terms*	First Hit Term Description	q-value
Coexpression	933	Rat Breast_Giusti09_300genes	1.72E-25
Coexpression Atlas	985	Heart muscle	8.01E-22
Computational	315	Neighborhood of MYL2	5.29E-22
Disease	969	Cardiomyopathy, Familial Idiopathic	1.47E-09
Drug	1522	Doxorubicin	2.88E-11
Gene Family	22	F-type ATPases Mitochondrial complex V: ATP synthase	2.28E-11
GO: BP	3311	Striated muscle contraction	2.01E-15
GO: CC	852	Sarcomere	2.53E-15
GO: MF	225	Structural constituent of muscle	8.08E-06
Human Phenotype	867	Sudden death	2.04E-07
Interaction	280	VDAC1 interactions	1.94E-05
MicroRNA	1	Hsa-miR-610:mirSVR_nonconserved_lowEffect-0.1-0.5	3.77E-05
Mouse Phenotype	1419	Cardiac hypertrophy	2.47E-09
Pathway	1340	Cardiac muscle contraction	4.01E-14
Pubmed	10774	Clinical features and outcome of hypertrophic cardiomyopathy associated with triple sarcomere protein gene mutations.	1.13E-19
ToppCell Atlas	209	MouseAtlas Mouse Multiple Adult muscle Subtype	2.90E-18
TFBS	21	Heart_cardiac muscle cell V\$MEF2_02	8.40E-06

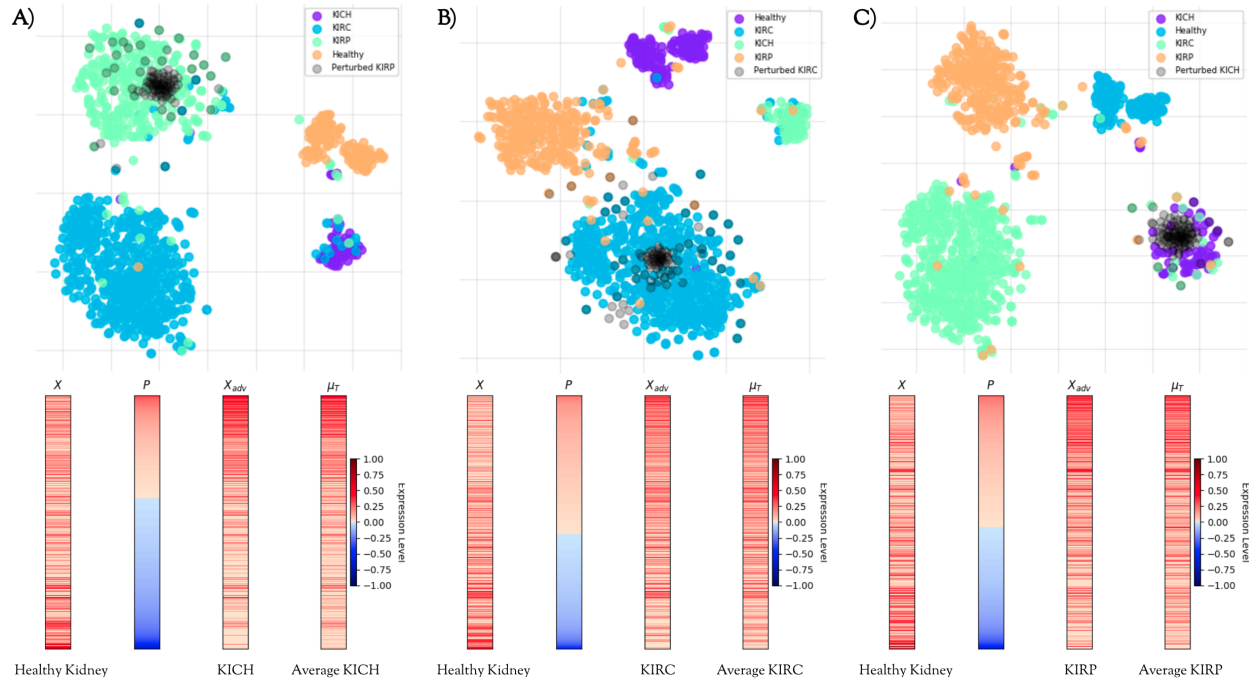


Figure 5: **Adversarial generation for three subtypes of Kidney cancer using all Hallmark genes as the input gene set.** t-SNE plot and corresponding heatmap of cellular transformation from healthy to KICH (A), healthy to KIRC (B), and healthy to KIRP (C). Perturbations ( $P$ ) range from  $[-1, 1]$ , which is added to original sample ( $X$ ), then adversarial example ( $X_{adv}$ ) is clipped to  $[0, 1]$ . The mean expression vector ( $\mu_T$ ) of the target class is shown.

In order to determine the efficacy of TSPG on more subtle transcriptome state transitions, we tested TSPG on samples from the same tissue of origin, namely healthy and cancerous kidney tissue extracted from the TCGA project. Using all 4,386 Hallmark genes as input, we examined the transcriptome transitions between healthy (“solid tissue normal”)

into one of three renal tumor subtypes: clear cell renal carcinoma (KIRC), renal papillary cell carcinoma (KIRP), and chromophobe renal cell carcinoma (KICH) as shown in Figure 5. For each transition between healthy and the three tumor sub-types, we identified the genes that were among the top 20 genes most positively and negatively perturbed during transition (Supplemental Table S2). For the most negatively perturbed genes in each case, the genes tended to be enriched for top expressed genes in kidney tissue as defined by the GenitoUrinary Development Molecular Anatomy Project (GUDMAP) database [32]. In contrast to the negatively perturbed genes appearing to have a normal kidney gene expression pattern, the top positively perturbed genes showed different enriched functions for each tumor subtype. A notable enriched function in KIRC genes was an association with genes involved in hypoxia, a hallmark of tumor progression where tumors avoid senescence by increasing blood supply (e.g. MSigDB hallmark hypoxia gene set was enriched). A notable function enriched in positively perturbed KIRP genes is an enrichment in membrane-binding annexin activity (e.g. [33]), a dysregulated function in cancer [34] including renal carcinoma [35, 17]. Finally, positively perturbed genes from healthy to KICH exhibited an enrichment for genes upregulated by the estrogen receptor alpha (e.g. [36]), and estrogen responsive genes have been implicated in kidney cancer [37]. These results suggest that negatively perturbed genes exhibit similar functions when transitioning to any RCC subtype, but the positively perturbed genes encode collective function unique to each subtype. Simply put, negatively perturbed genes are related to source tissue (i.e. healthy kidney) functionality, while positively perturbed genes relate to functionality specific to the target tissue (i.e. three cancerous kidney subtypes).

## 4 Discussion

Our preliminary experiments confirmed that neural networks used for classification of RNAseq data are indeed vulnerable to adversarial attacks. The results of Table 1 show that it is actually quite trivial for the generator to "trick" the target model into a specific classification no matter what the input is. This is likely due to the fact that the target model is relatively simple and was not trained using any defensive strategies against attacks. It is noteworthy that [15] found that the fundamental nature of adversarial examples do not differ across models trained on different subsets of data or models that had varying hyperparameters (number of layers, number of neurons in each layer, weight normalization schemes, etc.). Thus, we decided it was not necessary to test across different MLPs. Additionally, a noteworthy standard for adversarial attacks in image processing is that the perturbation is small in magnitude (equation 3); ideally, the perturbation to an input is not noticeable to the human eye. While this criterion is foundational for adversarial attacks in the computer vision domain, we are more interested in making *meaningful* transitions from source to target states, not in meaninglessly fooling a neural network. Therefore, despite including equation 3 to stabilize training, it was not imperative that the perturbations were small in magnitude. A large shift in gene expression may be required to transition between two disparate tissue states.

Due to the remarkable success of GANs in various data intensive applications, the objective of fooling a MLP was expected to be accomplished trivially. Therefore, we decided to take the adversarial attack paradigm several steps forward and demonstrated that generative adversarial networks produce biologically meaningful perturbations when trained under the proper criteria. We adapted and added criteria to the AdvGAN framework, which originated from within the computer vision community, to produce meaningful state transitions. While we used equation 3 to limit the magnitude of the perturbation, we found that adding equation 4 to the generator loss function significantly improved the realism of the results with respect to t-SNE plots and heatmap results.

What do these results mean for the study of biological systems? While the results of this analysis pointed to the near perfect adversarial attack rates (Table 1) and clear correspondence of the perturbed gene expression patterns to the target condition (Figure 2-5), we were concerned that we had nonsensically transformed biological states. However, the strong signal of biological function (Table 2, Table S2) and continued selection of the same positively perturbed genes leading to the target gene expression pattern suggests that the algorithm shifts the expression patterns in a biologically meaningful way. Thus, TSPG could be a novel method to detect genetic subsystems that are responsible for differentiating between both differing tissues as well as wild type and aberrant tissue states. Furthermore, generative models, particularly GANs, have been sparingly used in the field of biology. This can be attributed to a multitude of factors including the limited number of adequate datasets and the unnatural structure of RNAseq data when compared to images and language waveforms. This work serves as a proof of concept that GANs can be applied to RNAseq data to uncover meaningful gene expression patterns between different human tissue samples.

## 5 Acknowledgments

This work was supported by National Science Foundation Award #1659300 "CC\*Data: National Cyberinfrastructure for Scientific Data Analysis at Scale (SciDAS)". Clemson University is acknowledged for generous allotment of compute



time on the Palmetto cluster. We would also like to thank Sufeng Niu and Courtney Shearer for the valuable discussions we had regarding this work.

## References

- [1] Kimberly R Kukurba and Stephen B Montgomery. Rna sequencing and analysis. *Cold Spring Harbor Protocols*, 2015(11):pdb-top084970, 2015.
- [2] K. E. Roche, M. Weinstein, L. J. Dunwoodie, W. L. Poehlman, and F. A. Feltus. Sorting Five Human Tumor Types Reveals Specific Biomarkers and Background Classification Genes. *Sci Rep*, 8(1):8180, May 2018.
- [3] Yifei Chen, Yi Li, Rajiv Narayan, Aravind Subramanian, and Xiaohui Xie. Gene expression inference with deep learning. *Bioinformatics*, 32(12):1832–1839, 2016.
- [4] Chieh Lin, Siddhartha Jain, Hannah Kim, and Ziv Bar-Joseph. Using neural networks for reducing the dimensions of single-cell rna-seq data. *Nucleic acids research*, 45(17):e156–e156, 2017.
- [5] Yann LeCun, Yoshua Bengio, and Geoffrey Hinton. Deep learning. *nature*, 521(7553):436, 2015.
- [6] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. Imagenet classification with deep convolutional neural networks. In *Advances in neural information processing systems*, pages 1097–1105, 2012.
- [7] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778, 2016.
- [8] Geoffrey Hinton, Li Deng, Dong Yu, George Dahl, Abdel-rahman Mohamed, Navdeep Jaitly, Andrew Senior, Vincent Vanhoucke, Patrick Nguyen, Brian Kingsbury, et al. Deep neural networks for acoustic modeling in speech recognition. *IEEE Signal processing magazine*, 29, 2012.
- [9] Karl Moritz Hermann, Tomas Kocisky, Edward Grefenstette, Lasse Espeholt, Will Kay, Mustafa Suleyman, and Phil Blunsom. Teaching machines to read and comprehend. In *Advances in neural information processing systems*, pages 1693–1701, 2015.
- [10] Diederik P Kingma and Max Welling. Auto-encoding variational bayes. *arXiv preprint arXiv:1312.6114*, 2013.
- [11] Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative adversarial nets. In *Advances in neural information processing systems*, pages 2672–2680, 2014.
- [12] Phillip Isola, Jun-Yan Zhu, Tinghui Zhou, and Alexei A Efros. Image-to-image translation with conditional adversarial networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1125–1134, 2017.
- [13] Tero Karras, Timo Aila, Samuli Laine, and Jaakko Lehtinen. Progressive growing of gans for improved quality, stability, and variation. *arXiv preprint arXiv:1710.10196*, 2017.
- [14] Andrew Brock, Jeff Donahue, and Karen Simonyan. Large scale gan training for high fidelity natural image synthesis. *arXiv preprint arXiv:1809.11096*, 2018.
- [15] Christian Szegedy, Wojciech Zaremba, Ilya Sutskever, Joan Bruna, Dumitru Erhan, Ian Goodfellow, and Rob Fergus. Intriguing properties of neural networks. *arXiv preprint arXiv:1312.6199*, 2013.
- [16] Ian J Goodfellow, Jonathon Shlens, and Christian Szegedy. Explaining and harnessing adversarial examples. *arXiv preprint arXiv:1412.6572*, 2014.
- [17] Nicholas Carlini and David Wagner. Towards evaluating the robustness of neural networks. In *2017 IEEE Symposium on Security and Privacy (SP)*, pages 39–57. IEEE, 2017.
- [18] Yanpei Liu, Xinyun Chen, Chang Liu, and Dawn Song. Delving into transferable adversarial examples and black-box attacks. *arXiv preprint arXiv:1611.02770*, 2016.
- [19] Chaowei Xiao, Bo Li, Jun-Yan Zhu, Warren He, Mingyan Liu, and Dawn Song. Generating adversarial examples with adversarial networks. *arXiv preprint arXiv:1801.02610*, 2018.
- [20] Arsham Ghahramani, Fiona M Watt, and Nicholas M Luscombe. Generative adversarial networks uncover epidermal regulators and predict single cell perturbations. *bioRxiv*, page 262501, 2018.
- [21] Martin Arjovsky, Soumith Chintala, and Léon Bottou. Wasserstein gan. *arXiv preprint arXiv:1701.07875*, 2017.
- [22] Kamran Ghasedi Dizaji, Xiaoqian Wang, and Heng Huang. Semi-supervised generative adversarial network for gene expression inference. In *Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, pages 1435–1444. ACM, 2018.

- [23] J. N. Weinstein, E. A. Collisson, G. B. Mills, K. R. Shaw, B. A. Ozenberger, K. Ellrott, I. Shmulevich, C. Sander, J. M. Stuart, K. Chang, C. J. Creighton, C. Davis, L. Donehower, J. Drummond, D. Wheeler, A. Ally, M. Balasundaram, I. Birol, S. N. Butterfield, A. Chu, E. Chuah, H. J. Chun, N. Dhalla, R. Guin, M. Hirst, C. Hirst, R. A. Holt, S. J. Jones, D. Lee, H. I. Li, M. A. Marra, M. Mayo, R. A. Moore, A. J. Mungall, A. G. Robertson, J. E. Schein, P. Sipahimalani, A. Tam, N. Thiessen, R. J. Varhol, R. Beroukhim, A. S. Bhatt, A. N. Brooks, A. D. Cherniack, S. S. Freeman, S. B. Gabriel, E. Helman, J. Jung, M. Meyerson, A. I. Ojesina, C. S. Pedamallu, G. Saksena, S. E. Schumacher, B. Tabak, T. Zack, E. S. Lander, C. A. Bristow, A. Hadjipanayis, P. Haseley, R. Kucherlapati, S. Lee, E. Lee, L. J. Luquette, H. S. Mahadeshwar, A. Pantazi, M. Parfenov, P. J. Park, A. Protopopov, X. Ren, N. Santoso, J. Seidman, S. Seth, X. Song, J. Tang, R. Xi, A. W. Xu, L. Yang, D. Zeng, J. T. Auman, S. Balu, E. Buda, C. Fan, K. A. Hoadley, C. D. Jones, S. Meng, P. A. Mieczkowski, J. S. Parker, C. M. Perou, J. Roach, Y. Shi, G. O. Silva, D. Tan, U. Veluvolu, S. Waring, M. D. Wilkerson, J. Wu, W. Zhao, T. Bodenheimer, D. N. Hayes, A. P. Hoyle, S. R. Jeffreys, L. E. Mose, J. V. Simons, M. G. Soloway, S. B. Baylin, B. P. Berman, M. S. Bootwalla, L. Danilova, J. G. Herman, T. Hinoue, P. W. Laird, S. K. Rhie, H. Shen, T. Triche, D. J. Weisenberger, S. L. Carter, K. Cibulskis, L. Chin, J. Zhang, G. Getz, C. Sougnez, M. Wang, G. Saksena, S. L. Carter, K. Cibulskis, L. Chin, J. Zhang, G. Getz, H. Dinh, H. V. Doddapaneni, R. Gibbs, P. Gunaratne, Y. Han, D. Kalra, C. Kovar, L. Lewis, M. Morgan, D. Morton, D. Muzny, J. Reid, L. Xi, J. Cho, D. DiCara, S. Frazer, N. Gehlenborg, D. I. Heiman, J. Kim, M. S. Lawrence, P. Lin, Y. Liu, M. S. Noble, P. Stojanov, D. Voet, H. Zhang, L. Zou, C. Stewart, B. Bernard, R. Bressler, A. Eakin, L. Iype, T. Knijnenburg, R. Kramer, R. Kreisberg, K. Leinonen, J. Lin, Y. Liu, M. Miller, S. M. Reynolds, H. Rovira, I. Shmulevich, V. Thorsson, D. Yang, W. Zhang, S. Amin, C. J. Wu, C. C. Wu, R. Akbani, K. Aldape, K. A. Baggerly, B. Broom, T. D. Casasent, J. Cleland, C. Creighton, D. Dodda, M. Edgerton, L. Han, S. M. Herbrich, Z. Ju, H. Kim, S. Lerner, J. Li, H. Liang, W. Liu, P. L. Lorenzi, Y. Lu, J. Melott, G. B. Mills, L. Nguyen, X. Su, R. Verhaak, W. Wang, J. N. Weinstein, A. Wong, Y. Yang, J. Yao, R. Yao, K. Yoshihara, Y. Yuan, A. K. Yung, N. Zhang, S. Zheng, M. Ryan, D. W. Kane, B. A. Aksoy, G. Ciriello, G. Dresdner, J. Gao, B. Gross, A. Jacobsen, A. Kahles, M. Ladanyi, W. Lee, K. V. Lehmann, M. L. Miller, R. Ramirez, G. Ratsch, B. Reva, C. Sander, N. Schultz, Y. Senbabaoglu, R. Shen, R. Sinha, S. O. Sumer, Y. Sun, B. S. Taylor, N. Weinhold, S. Fei, P. Spellman, C. Benz, D. Carlin, M. Cline, B. Craft, K. Ellrott, M. Goldman, D. Haussler, S. Ma, S. Ng, E. Paull, A. Radenbaugh, S. Salama, A. Sokolov, J. M. Stuart, T. Swatloski, V. Uzunangelov, P. Waltman, C. Yau, J. Zhu, S. R. Hamilton, G. Getz, C. Sougnez, S. Abbott, R. Abbott, N. D. Dees, K. Delehaunty, L. Ding, D. J. Dooling, J. M. Eldred, C. C. Fronick, R. Fulton, L. L. Fulton, J. Kalicki-Veizer, K. L. Kanchi, C. Kandoth, D. C. Koboldt, D. E. Larson, T. J. Ley, L. Lin, C. Lu, V. J. Magrini, E. R. Mardis, M. D. McLellan, J. F. McMichael, C. A. Miller, M. O’Laughlin, C. Pohl, H. Schmidt, S. M. Smith, J. Walker, J. W. Wallis, M. C. Wendl, R. K. Wilson, T. Wylie, Q. Zhang, R. Burton, M. A. Jensen, A. Kahn, T. Pihl, D. Pot, Y. Wan, D. A. Levine, A. D. Black, J. Bowen, J. Frick, J. M. Gastier-Foster, H. A. Harper, C. Helsel, K. M. Leraas, T. M. Lichtenberg, C. McAllister, N. C. Ramirez, S. Sharpe, L. Wise, E. Zmuda, S. J. Chanock, T. Davidsen, J. A. Demchok, G. Eley, I. Felau, B. A. Ozenberger, M. Sheth, H. Sofia, L. Staudt, R. Tarnuzzer, Z. Wang, L. Yang, J. Zhang, L. Omberg, A. Margolin, B. J. Raphael, F. Vandin, H. T. Wu, M. D. Leiserson, S. C. Benz, C. J. Vaske, H. Noushmehr, T. Knijnenburg, D. Wolf, L. Van ’t Veer, E. A. Collisson, D. Anastassiou, T. H. Ou Yang, N. Lopez-Bigas, A. Gonzalez-Perez, D. Tamborero, Z. Xia, W. Li, D. Y. Cho, T. Przytycka, M. Hamilton, S. McGuire, S. Nelander, P. Johansson, R. Jornsten, T. Kling, and J. Sanchez. The Cancer Genome Atlas Pan-Cancer analysis project. *Nat. Genet.*, 45(10):1113–1120, Oct 2013.
- [24] Aravind Subramanian, Pablo Tamayo, Vamsi K Mootha, Sayan Mukherjee, Benjamin L Ebert, Michael A Gillette, Amanda Paulovich, Scott L Pomeroy, Todd R Golub, Eric S Lander, et al. Gene set enrichment analysis: a knowledge-based approach for interpreting genome-wide expression profiles. *Proceedings of the National Academy of Sciences*, 102(43):15545–15550, 2005.
- [25] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay. Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research*, 12:2825–2830, 2011.
- [26] Martín Abadi, Ashish Agarwal, Paul Barham, Eugene Brevdo, Zhifeng Chen, Craig Citro, Greg S. Corrado, Andy Davis, Jeffrey Dean, Matthieu Devin, Sanjay Ghemawat, Ian Goodfellow, Andrew Harp, Geoffrey Irving, Michael Isard, Yangqing Jia, Rafal Jozefowicz, Lukasz Kaiser, Manjunath Kudlur, Josh Levenberg, Dandelion Mané, Rajat Monga, Sherry Moore, Derek Murray, Chris Olah, Mike Schuster, Jonathon Shlens, Benoit Steiner, Ilya Sutskever, Kunal Talwar, Paul Tucker, Vincent Vanhoucke, Vijay Vasudevan, Fernanda Viégas, Oriol Vinyals, Pete Warden, Martin Wattenberg, Martin Wicke, Yuan Yu, and Xiaoqiang Zheng. TensorFlow: Large-scale machine learning on heterogeneous systems, 2015. Software available from tensorflow.org.
- [27] Xudong Mao, Qing Li, Haoran Xie, Raymond YK Lau, Zhen Wang, and Stephen Paul Smolley. Least squares generative adversarial networks. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 2794–2802, 2017.

- [28] Jun-Yan Zhu, Taesung Park, Phillip Isola, and Alexei A Efros. Unpaired image-to-image translation using cycle-consistent adversarial networks. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 2223–2232, 2017.
- [29] Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014.
- [30] Laurens van der Maaten and Geoffrey Hinton. Visualizing data using t-sne. *Journal of machine learning research*, 9(Nov):2579–2605, 2008.
- [31] Jing Chen, Eric E Bardes, Bruce J Aronow, and Anil G Jegga. Toppgene suite for gene list enrichment analysis and candidate gene prioritization. *Nucleic acids research*, 37(suppl\_2):W305–W311, 2009.
- [32] Simon D Harding, Chris Armit, Jane Armstrong, Jane Brennan, Ying Cheng, Bernard Haggarty, Derek Houghton, Sue Lloyd-MacGilp, Xingjun Pi, Yogmatee Roochun, et al. The gudmap database—an online resource for genitourinary research. *Development*, 138(13):2845–2853, 2011.
- [33] Ursula Rescher and Volker Gerke. Annexins—unique membrane binding proteins with diverse functions. *Journal of cell science*, 117(13):2631–2639, 2004.
- [34] Stine Prehn Lauritzen, Theresa Louise Boye, and Jesper Nylandsted. Annexins are instrumental for efficient plasma membrane repair in cancer cells. In *Seminars in cell & developmental biology*, volume 45, pages 32–38. Elsevier, 2015.
- [35] Shun-Fa Yang, Han-Lin Hsu, Tai-Kuang Chao, Chia-Jung Hsiao, Yung-Feng Lin, and Chao-Wen Cheng. Annexin a2 in renal cell carcinoma: expression, function, and prognostic significance. In *Urologic Oncology: Seminars and Original Investigations*, volume 33, pages 22–e11. Elsevier, 2015.
- [36] Rebecca A Stein, Ching-yi Chang, Dmitri A Kazmin, James Way, Thies Schroeder, Melanie Wergin, Mark W Dewhirst, and Donald P McDonnell. Estrogen-related receptor  $\alpha$  is critical for the growth of estrogen receptor-negative breast cancer. *Cancer research*, 68(21):8805–8812, 2008.
- [37] Zhihong Liu, You Lu, Zonghai He, Libo Chen, and Yiping Lu. Expression analysis of the estrogen receptor target genes in renal cell carcinoma. *Molecular medicine reports*, 11(1):75–82, 2015.